

Introduction

Machine Learning:
Chapter 1
Advanced AI Course

Copyright (c) School of Computer Sci. & Eng. SNU

목 차

- Definition and Applications of Machine
- Designing a Learning System
 - ◆ Choosing the Training Experience
 - ◆ Choosing the Target Function
 - ◆ Choosing a Representation for the Target Function
 - ◆ Choosing a Function Approximation Algorithm
 - ◆ The Final Design
- Perspectives and Issues in Machine Learning
- Organization of the Book
- Summary and Bibliography

Copyright (c) School of Computer Sci. & Eng. SNU

Applications of Machine Learning

- Recognizing spoken words
 - ◆ 음소와 단어 인식, 신호 해석
 - ◆ 신경망, Hidden Markov models
- Driving an autonomous vehicle
 - ◆ 무인 자동차 운전, 센서 기반 제어 등에도 응용
- Classifying new astronomical structures
 - ◆ 천체 물체 분류, Decision tree learning 기법 사용
- Playing world-class Backgammon
 - ◆ 실제 게임을 통해서 전략을 학습, 탐색공간 문제에 응용

Copyright (c) School of Computer Sci. & Eng. SNU

Disciplines Related with Machine Learning

- Artificial intelligence
 - ◆ 기호 표현 학습, 탐색문제, 문제해결, 기존지식의 활용
- Bayesian methods
 - ◆ 가설 확률계산의 기초, naïve Bayes classifier, unobserved 변수 추정
- Computational complexity theory
 - ◆ 계산 효율, 학습 데이터의 크기, 오류의 수 등의 측정에 필요한 이론적 기반
- Control theory
 - ◆ 이미 정의된 목적을 최적화하는 제어과정과 다음 상태 예측을 학습

Copyright (c) School of Computer Sci. & Eng. SNU

Disciplines Related with Machine Learning (2)

- Information theory
 - ◆ Entropy와 Information Content를 측정, Minimum Description Length, Optimal Code와 Optimal Training의 관계
- Philosophy
 - ◆ Occam's Razor, 일반화의 타당성 분석
- Psychology and neurobiology
 - ◆ Neural network models
- Statistics
 - ◆ 가설의 정확도 추정시 발생하는 에러의 특성화, 신뢰구간, 통계적 검증

Copyright (c) School of Computer Sci. & Eng. SNU

Well-posed Learning Problems

- Definition
 - ◆ A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .
- A class of tasks T
- Experience E
- Performance measure P

Copyright (c) School of Computer Sci. & Eng. SNU

A Checkers Learning Problem

- Three Features: 학습문제의 정의
 - ♦ The class of tasks
 - ♦ The measure of performance to be improved
 - ♦ The source of experience
- Example
 - ♦ Task T : playing checkers
 - ♦ Performance measure P : percent of games won against opponent
 - ♦ Training experience E : playing practice games against itself

Copyright (c) School of Computer Sci. & Eng. SNU

1.2 Designing a Learning System

- Choosing the Training Experience
- Choosing the Target Function
- Choosing a Representation for the Target Function
- Choosing a Function Approximation Algorithm

Copyright (c) School of Computer Sci. & Eng. SNU

Choosing the Training Experience

- Key Attributes
 - ♦ Direct/indirect feedback
 - Direct feedback: checkers state and correct move
 - Indirect feedback: move sequence and final outcomes
 - ♦ Degree of controlling the sequence of training example
 - Learner가 학습 정보를 얻을 때 teacher의 도움을 받는 정도
 - ♦ Distribution of examples
 - 시스템의 성능을 평가하는 테스트의 예제 분포를 잘 반영해야 함

Copyright (c) School of Computer Sci. & Eng. SNU

Choosing the Target Function

- A function that chooses the best move M for any B
 - ♦ $ChooseMove : B \rightarrow M$
 - ♦ Difficult to learn
- It is useful to reduce the problem of improving performance P at task T to the problem of learning some particular *target function*.
- An evaluation function that assigns a numerical score to any B
 - ♦ $V : B \rightarrow R$

Copyright (c) School of Computer Sci. & Eng. SNU

Target Function for the Checkers Problem

- Algorithm
 - ♦ If b is a final state that is won, then $V(b) = 100$
 - ♦ that is lost, then $V(b) = -100$
 - ♦ that is drawn, then $V(b) = 0$
 - ♦ If b is not a final state, then $V(b) = V(b')$, where b' is the best final board state
- Nonoperational, i.e. not efficiently computable definition
- Operational description of V needs function approximation

Copyright (c) School of Computer Sci. & Eng. SNU

Choosing a Representation for the Target Function

- Describing the function
 - ♦ Tables
 - ♦ Rules
 - ♦ Polynomial functions
 - ♦ Neural nets
- Trade-off in choice
 - ♦ Expressive power
 - ♦ Size of training data

Copyright (c) School of Computer Sci. & Eng. SNU

Linear Combination as Representation

$$(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$$

- x_1 : # of black pieces on the board
- x_2 : # of red pieces on the board
- x_3 : # of black kings on the board
- x_4 : # of red kings on the board
- x_5 : # of black pieces threatened by red
- x_6 : # of red pieces threatened by black
- $w_1 - w_6$: weights

Copyright (c) School of Computer Sci. & Eng. SNU

Partial Design of a Checkers Learning Program

- Task T : playing checkers
- Performance measure P : Percent of games won in the world tournament
- Training experience E : games played against itself
- Target function V : $Board \rightarrow R$
- Target function representation
 - $(b) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6$

Copyright (c) School of Computer Sci. & Eng. SNU

Choosing a Function Approximation Algorithm

- A training example is represented as an ordered pair $\langle b, V_{train}(b) \rangle$
 - ♦ b : board state
 - ♦ $V_{train}(b)$: training value for b
- Instance: "black has won the game ($x_2 = 0$)"
 - ♦ $\langle x_1=3, x_2=0, x_3=1, x_4=0, x_5=0, x_6=0 \rangle, +100$
- Estimating training values for intermediate board states
 - ♦ $V_{train}(b) \leftarrow (Successor(b))$
 - ♦ V : current approximation to V
 - ♦ $Successor(b)$: the next board state

Copyright (c) School of Computer Sci. & Eng. SNU

Adjusting the Weights

- Choosing w_i to best fit the training examples
- Minimize the squared error
- LMS Weight Update Rule
 - For each training example $\langle b, V_{train}(b) \rangle$
 1. Use the current weights to calculate $V'(b)$
 2. For each weight w_i , update it as

Copyright (c) School of Computer Sci. & Eng. SNU

The Final Design

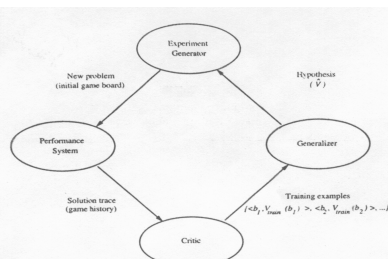


FIGURE 1.1
Final design of the checkers learning program.

Copyright (c) School of Computer Sci. & Eng. SNU

Four Components of a Learning System

- Performance system
 - ♦ Solve the given performance task
 - ♦ Use the learned target function
 - ♦ New problem \rightarrow trace of its solution
- Critic
 - ♦ Output a set of training examples of the target function

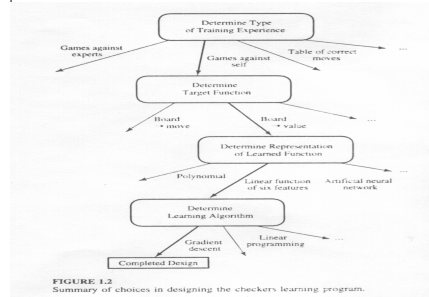
Copyright (c) School of Computer Sci. & Eng. SNU

Four Components of a Learning System (2)

- Generalizer
 - ◆ Input: training example
 - ◆ Output: hypothesis (estimate of the target function)
 - ◆ Generalizes from the specific training examples
 - ◆ Hypothesizes a general function
- Experiment generator
 - ◆ Input - current hypothesis
 - ◆ Output - a new problem
 - ◆ Picks new practice problem maximizing the learning rate

Copyright (c) School of Computer Sci. & Eng. SNU

Sequence of Design Choices



Copyright (c) School of Computer Sci. & Eng. SNU

Alternative Algorithms for Checkers Learning

- Nearest neighbor algorithms (Chap. 8)
- Genetic algorithms (Chap. 9)
- Explanation-based learning (Chap. 11)

Copyright (c) School of Computer Sci. & Eng. SNU

1.3 Perspectives and Issues in ML

- “Learning as *search* in a space of possible hypotheses”
- Representations for hypotheses
 - ◆ Linear functions
 - ◆ Logical descriptions
 - ◆ Decision trees
 - ◆ Neural networks
- Learning methods are characterized by their search strategies and by the underlying structure of the search spaces.

Copyright (c) School of Computer Sci. & Eng. SNU

Issues in Machine Learning

- Algorithm for learning general target function from specific training examples
- Amount of data
- Helpful prior knowledge
- Choice of strategy for next training experience
- Method of reducing the learning task to function approximation
- Automatically alter its representation for target function

Copyright (c) School of Computer Sci. & Eng. SNU

Questions about Machine Learning

- What **algorithms** exist for learning general target functions from specific examples?
 - ◆ In what settings will particular algorithms **converge** to the desired function, given sufficient training data?
 - ◆ Which algorithms perform best for which types of problems and representations?
- How much **training data** is sufficient?
 - ◆ What general bound can be found to relate the confidence in learned hypotheses to the amount of training experience and the character of the learner’s hypothesis space?

Copyright (c) School of Computer Sci. & Eng. SNU

Questions (2)

- When and how can **prior knowledge** held by the learner **guide the process of generalizing** from examples?
 - ◆ Can prior knowledge be helpful even when it is only approximately correct?
- What is the best strategy for **choosing a useful next training experience**, and how does the choice of this strategy alter the complexity of the learning problem?

Copyright (c) School of Computer Sci. & Eng. SNU

Questions (3)

- What is the best way to **reduce the learning task to one or more function approximation problems**?
 - ◆ What specific functions should the system attempt to learn?
 - ◆ Can this process itself be automated?
- How can the learner **automatically alter its representation** to improve its ability to represent and learn the target function?

Copyright (c) School of Computer Sci. & Eng. SNU

요약

- 기계학습은 다양한 응용분야에서 실용적 가치가 크다.
 - ◆ 많은 데이터로부터 규칙성을 발견하는 문제(data mining)
 - ◆ 문제의 성격 규명이 어려워 효과적인 알고리즘을 개발할 지식이 없는 문제 영역(human face recognition)
 - ◆ 변화하는 환경에 동적으로 적응하여야 하는 문제 영역(manufacturing process control)
- 기계학습은 다양한 다른 학문 분야와 밀접히 관련된다.
 - ◆ 인공지능, 확률통계, 정보이론, 계산이론, 심리학, 신경과학, 제어이론, 철학
- 잘 정의된 학습 문제는 다음을 요구한다.
 - ◆ 문제(task)의 명확한 기술, 성능평가 기준, 훈련경험을 위한

Copyright (c) School of Computer Sci. & Eng. SNU

요약 (계속)

- 기계학습 시스템의 설계 시에는 다음 사항을 고려하여야 한다.
 - ◆ 훈련경험의 유형 선택
 - ◆ 학습할 목표함수
 - ◆ 목표함수에 대한 표현
 - ◆ 훈련 예로부터 목표함수를 학습하기 위한 알고리즘
- 학습은 가능한 가설 공간에서 주어진 훈련 예와 다른 배경지식을 가장 잘 반영하는 하나의 가설을 탐색하는 탐색이다.
 - ◆ 다양한 학습 방법은 서로 다른 가설공간의 형태와 이 공간 내에서 탐색을 수행하는 전략에 의해 규정 지어진다.

Copyright (c) School of Computer Sci. & Eng. SNU